#670 **Preordine**

Se dă un arbore binar care conține valori numere naturale. Să se afișeze valorile din arbore în urma parcurgerii în preordine.

#include <fstream>

using namespace std;

int st[1001],dr[1001],n,v[1001];

ifstream fin("preordine.in");

ofstream fout("preordine.out");

void preord(int r)

{

if(r)

{

fout<<v[r]<<' ';

preord(st[r]);

preord(dr[r]);

}

}

int main()

{

int i,s=0,r;

fin>>n;

for(i=1;i<=n;i++)

{

fin>>v[i]>>st[i]>>dr[i];

s=s+st[i]+dr[i];

}

if(n%2==0)r=n/2\*(n+1);

else r=(n+1)/2\*n;

r=r-s;

preord(r);

return 0;

}

**#671 Inordine**

#include <fstream>

using namespace std;

ifstream fin("inordine.in");

ofstream fout("inordine.out");

int st[1001],dr[1001],x[1001];

int r,s;

void SRD(int r)

{

if(r!=0)

{

SRD(st[r]);

fout<<x[r]<<' ';

SRD(dr[r]);

}

}

int main()

{

int n,i;

fin>>n;

for(i=1; i<=n; i++)

{

fin>>x[i]>>st[i]>>dr[i];

s=s+st[i]+dr[i];

}

r=n\*(n+1)/2-s ;

SRD(r);

return 0;

}

**672 Postordine**

#include <fstream>

using namespace std;

ifstream fin("postordine.in");

ofstream fout("postordine.out");

int st[1001],dr[1001],x[1001];

int r,s;

void SDR(int r)

{

if(r!=0)

{

SDR(st[r]);

SDR(dr[r]);

fout<<x[r]<<' ';

}

}

int main()

{

int n,i;

fin>>n;

for(i=1; i<=n; i++)

{

fin>>x[i]>>st[i]>>dr[i];

s=s+st[i]+dr[i];

}

r=n\*(n+1)/2-s ;

SDR(r);

return 0;

}